Symlink

Vulnerable to TOCTOU issues

Sean Barnum, Cigital, Inc. [vita¹]

Copyright © 2007 Cigital, Inc.

2007-04-24

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 5310 bytes

Attack Category	• Path spoofing or confusion problem		
Vulnerability Category	Indeterminate File/PathTOCTOU - Time of Check, Time of Use		
Software Context	File Management		
Location			
Description	The symlink() function creates a symbolic link. It is generally vulnerable to classic TOCTOU attacks.		
	A call to symlink() should be flagged if the first or second argument is used earlier in a check-category call.		
APIs	Function Name Comments		
	symlink		
Method of Attack	The key issue with respect to TOCTOU vulnerabilities is that programs make assumptions about atomicity of actions. It is assumed that checking the state or identity of a targeted resource followed by an action on that resource is all one action. In reality, there is a period of time between the check and the use that allows either an attacker to intentionally or another interleaved process or thread to unintentionally change the state of the targeted resource and yield unexpected and undesired results.		
	The symlink() call is a use-category call, which when preceded by a check-category call can be indicative of a TOCTOU vulnerability.		
	A TOCTOU attack in regards to symlink() can occur when		
	a. A check for the existence of the file occurs or a non-fd reference (pathname) to the filename occurs		
	b. An actual call to symlink occurs.		
	Between a and b, an attacker could, for example, link the referenced file to a known file. The subsequent		

^{1.} http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

Symlink 1

	symlink() call w impact.	symlink() call would have an unintended effect or impact.		
Exception Criteria				
Solutions	Solution Applicability	Solution Description	Solution Efficacy	
	Generally applicable to any symlink.	Utilize a file descriptor version of stat/ fstat when checking.	Effective.	
	Generally applicable to any symlink.	The most basic advice for TOCTOU vulnerabilities is to not perform a check before the use. This does not resolve the underlying issue of the execution of a function on a resource whose state and identity cannot be assured, but it does help to limit the false sense of security given by the check.	Does not resolve the underlying vulnerability but limits the false sense of security given by the check.	
	Generally applicable to any symlink.	Limit the interleaving of operations on files from multiple processes.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.	

Symlink

Generally

applicable to

any symlink.

Limit the spread Does not

eliminate the

underlying vulnerability

but can help

make it more difficult to exploit.

2

of time (cycles)

between the

check and use

of a resource.

	applicable to any symlink. t	resource after the use call to verify that the action was taken appropriately.	status after the operation does not change the fact that the operation may have been exploited but it does allow halting of the application in an error state to help limit further damage.	
Signature Details	int symlink (const of *newname);	int symlink (const char *oldname, const char *newname);		
Examples of Incorrect Code	int use_state struct state check_status: &statbuf);	<pre>use_status=symlink(oldname, newname);</pre>		
Examples of Corrected Code	0)	<pre>if (symlink(oldname, newname) != 0) printf("Error creating link\n");</pre>		
Source Reference	• ITS4 Source C	ITS4 Source Code Vulnerability Scanning Tool 2		
Recommended Resources				
Discriminant Set	Operating System	n • U	NIX (All)	
	Languages	• C	++	

Generally

Recheck the

Checking the

Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

Symlink 3

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about "Fair Use," contact Cigital at copyright@cigital.com¹. The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content. mailto:copyright@cigital.com

Symlink 4